10

O‘NINCHI AMALIY MASHG‘ULOT

Korxona ilovalarini dasturlash · 2 soat · 10-mavzu

Fon vazifalari: navbatlar va rejalashtirish

Jamoa loyihasi bosqichi · 5 ta PHP topshiriq · o‘qituvchi bahosi

Queues | Redis | Scheduler | Idempotentlik | Notifications

MASHG‘ULOT MAQSADI

Amaliy mashg‘ulot oxirida siz quyidagilarni bajara olasiz...

1. **Sozlash** — Redis navbatini, alohida worker va scheduler konteynerlarini hamda Mailpit ni
2. **Yozish** — GenerateSalesReport jobini va hisobotni 202 bilan qabul qiluvchi endpointni
3. **Qo‘llash** — tries, backoff, timeout va failed() ni haqiqiy joblarda
4. **Loyihalash** — kunda bir marta ishlaydigan idempotent SendLowStockAlert va LowStockNotification ni
5. **Rejalashtirish** — kam qoldiq tekshiruvini ish kunlari 09:00 (Asia/Tashkent) ga

Qanday bajariladi va qanday baholanadi?

Ish tartibi

1. Saytga login va parolingiz bilan kiring: **emro.uz/korxona/kirish.html**
2. **10-amaliy** sahifasini oching (oxirgi slayddagi QR kod).
3. Yo‘riqnoma bo‘yicha jamoa loyihasida bosqichni bajaring, **commit** va **push** qiling.
4. PHP topshiriqlarni yeching: ▶ **Sinab ko‘rish** → ✓ **Topshirish** (yashirin testlar).
5. Loyiha havolasini (GitHub repo, PR yoki tag) topshiring — o‘qituvchi **baho** qo‘yadi.

Baholash mezonlari

- **Natija** — loyiha bosqichi mezonlar ro‘yxati bo‘yicha bajarilgan.
- **Topshiriqlar** — PHP topshiriqlar testlardan o‘tgan.
- **Kod sifati** — tushunarli nomlar, PSR-12, kichik va mazmunli commitlar.
- **Jamoaviy hissa** — har bir a’zo o‘z qismini tushuntira oladi.
- **Muddat** — muddatdan keyin topshirilgan ish “kechikkan” deb belgilanadi.

01

Muhit: Redis navbati, worker va scheduler

docker-compose · .env · Mailpit · tekshiruv buyruqlari

Docker

Redis

Mailpit

DOCKER COMPOSE

queue, scheduler va mailpit servislari

```yaml
queue:
  build: .
  command: >
    php artisan queue:work redis
    --queue=reports,default --tries=3 --max-time=3600
  depends_on: [db, redis]
  restart: unless-stopped
scheduler:
  build: .
  command: php artisan schedule:work
  depends_on: [db, redis]
  restart: unless-stopped
mailpit:
  image: axllent/mailpit
  ports: ["8025:8025", "1025:1025"]
```

▶ NATIJA

```
$ docker compose ps
omborpro-queue-1       running
omborpro-scheduler-1   running
omborpro-mailpit-1     running   0.0.0.0:8025->8025/tcp
```

`build: .`

worker va scheduler — `app` bilan bir xil kod va obraz

`--queue=reports,default`

hisobotlar ustuvor; e-mail lar `default` da

`schedule:work`

har daqiqada `schedule:run` — faqat bitta replika!

mailpit 8025

barcha xatlarni ushlaydigan veb-interfeys

.ENV

Navbat, kesh, pochta va vaqt zonasi

```
APP_TIMEZONE=Asia/Tashkent
QUEUE_CONNECTION=redis
CACHE_STORE=redis
REDIS_HOST=redis
MAIL_MAILER=smtp
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_FROM_ADDRESS="noreply@omborpro.test"
MAIL_FROM_NAME="OmborPro"
```

▶ NATIJA

```
$ php artisan about --only=drivers
  Cache ............. redis
  Mail .............. smtp
  Queue ............. redis
```

CACHE_STORE=redis

ShouldBeUnique, onOneServer va queue:restart signali uchun

MAIL_HOST=mailpit

xatlar real manzillarga ketmaydi

APP_TIMEZONE

scheduler va `now()` Toshkent vaqtida

TEKSHIRUV

Muhit tayyorligini tekshirish

1

Migratsiya

`php artisan make:notifications-table` → `php artisan migrate`

2

Konteynerlar

`docker compose up -d` → `docker compose ps` — hammasi running

3

Worker logi

`docker compose logs -f queue` — “Processing jobs from …”

4

Sinov jobi

tinker: `App\Jobs\SendLowStockAlert::dispatch()` → logda DONE

5

Pochta

`http://localhost:8025` — Mailpit da xat ko‘rinadi

02

Hisobotni asinxron shakllantirish: GenerateSalesReport

ReportService::exportSales · job · endpoint · yuklab olish

GenerateSalesReport | 202 | Excel::store

REPORTSERVICE

exportSales: faylni diskka yozish

```php
public function exportSales(
    User $user, string $from, string $to,
): string {
    $path = "reports/{$user->id}/sales-{$from}_{$to}.xlsx";

    Excel::store(new SalesReportExport($from, $to), $path, 'local');

    return $path;
}
```

▶ NATIJA

```
storage/app/reports/7/sales-2026-10-01_2026-10-31.xlsx  (412 KB)
```

`$path`

aniq nom: takroriy bajarilish faylni qayta yozadi (idempotent)

`{$user->id}`

fayl egasi yo‘ldan aniq — yuklab olishda tekshiriladi

`Excel::store`

`maatwebsite/excel`: eksport sinfini diskka yozadi

JOB

GenerateSalesReport: to‘liq sozlamalar bilan

```php
class GenerateSalesReport implements ShouldQueue
{
    use Queueable;
    public int $tries = 3;
    public int $timeout = 80;
    public array $backoff = [30, 120];
    public function __construct(
        public User $user, public string $from, public string $to,
    ) {}

    public function handle(ReportService $reports): void
    {
        $path = $reports->exportSales(
            $this->user, $this->from, $this->to,
        );
        $this->user->notify(new ReportReadyNotification($path));
    }
}
```

▶ NATIJA

```
2026-10-27 10:15:02 App\Jobs\GenerateSalesReport ..... RUNNING
2026-10-27 10:15:06 App\Jobs\GenerateSalesReport . 3.91s DONE
```

`$timeout = 80`

`retry_after = 90` dan kichik

`$backoff = [30, 120]`

baza band bo‘lsa — 30 s, keyin 2 daqiqa kutish

ReportReadyNotification

faqat `database` kanali: qo‘ng‘iroqchada havola

SINOV SSENARIYLARI

Hisobot oqimini qo‘lda tekshirish

Ssenariy	Kutilgan natija	Qayerda ko‘rinadi
To‘g‘ri davr bilan so‘rov	`202`, javob < 100 ms	DevTools → Network
`to < from`	`422`, job navbatga tushmaydi	Javob JSON, worker logi bo‘sh
`omborchi` **roli bilan so‘rov**	`403`	Policy / Gate
Worker ishlayapti	RUNNING → DONE, fayl paydo bo‘ladi	`logs -f queue`, `storage/app/reports`
Worker to‘xtatilgan	Job Redis da kutadi, ishga tushgach bajariladi	`redis-cli LLEN ...queues:reports`
Begona foydalanuvchi faylni so‘raydi	`403` yoki `404`	`GET /reports/{file}`

03

Kam qoldiq bildirishnomasi va rejalashtirish

lowStock so‘rovi · SendLowStockAlert · LowStockNotification · Schedule

SendLowStockAlert

LowStockNotification

Schedule

REPOSITORY

ProductRepository::lowStock()

```php
public function lowStock(): Collection
{
    $sum = 'COALESCE(SUM(stocks.quantity - stocks.reserved), 0)';

    return Product::query()
        ->select('products.sku', 'products.name', 'products.min_stock')
        ->selectRaw("$sum AS available")
        ->leftJoin('stocks', 'stocks.product_id', '=', 'products.id')
        ->where('products.is_active', true)
        ->groupBy('products.id', 'products.sku', 'products.name',
            'products.min_stock')
        ->havingRaw("$sum < products.min_stock")
        ->orderBy('products.sku')
        ->get();
}
```

▶ NATIJA

```
sku      | name              | min_stock | available
UN-001   | Un 1-nav 50 kg    | 10        | 4
UN-014   | Shakar 50 kg      | 5         | 0
```

`quantity - reserved`

mavjud qoldiq — rezerv hisobga olinadi

`leftJoin`

ombor yozuvi umuman yo‘q mahsulot ham chiqadi (COALESCE → 0)

`havingRaw`

agregat bo‘yicha filtr — `WHERE` da emas

TINKER BILAN SINOV

Idempotentlikni qo‘lda tekshirish

```
php artisan tinker
> Stock::where('product_id', 1)->update(['quantity' => 3]);
> App\Jobs\SendLowStockAlert::dispatchSync();
> App\Jobs\SendLowStockAlert::dispatchSync();
> Cache::get('low-stock-alert:' . now()->toDateString());
> exit

php artisan schedule:list
php artisan schedule:test --name="App\Jobs\SendLowStockAlert"
```

▶ NATIJA

```
= true
  0 9 * * 1-5  App\Jobs\SendLowStockAlert ... Next Due: 21 hours
from now
Mailpit: 1 ta xat (“Kam qoldiq: 1 ta mahsulot”)
```

`dispatchSync()`

navbatsiz darhol bajarish — tez sinov uchun

ikki marta chaqirish

ikkinchisi `Cache::has` da to‘xtaydi — xat bitta

`schedule:test`

jadvaldagi vazifani vaqtini kutmasdan ishga tushiradi

TEKSHIRUV

Yaxshi va yomon fon vazifasi

✗ Noto‘g‘ri

- Har chaqiriqda xat — retry da omborchilar 5 ta xat oladi
- Kolleksiyani bildirishnomaga berish — `available` yo‘qoladi
- Vaqt zonasisiz `dailyAt('09:00')` — xat 14:00 da
- `failed()` yo‘q — xato jim o‘tib ketadi

✓ To‘g‘ri

- Kunlik belgi + `ShouldBeUnique` — kuniga bitta xat
- Oddiy massiv: `sku`, `name`, `available`, `min_stock`
- `->timezone('Asia/Tashkent')->onOneServer()`
- `failed()` logga yozadi va adminni ogohlantiradi

LOYIHA BOSQICHI

Natija va topshirish

Natija: hisobot so‘rovi 202 javob bilan darhol qaytadi va worker tomonidan fon rejimida tayyorlanadi; ish kunlari 09:00 da kam qoldiq haqida e-mail va panel bildirishnomasi avtomatik, kuniga bir marta yuboriladi.

Baholash mezonlari

✓ `docker-compose.yml` da `queue`, `scheduler` va `mailpit` servislari bor; `QUEUE_CONNECTION=redis`.

✓ `POST /reports/sales` 202 qaytaradi, noto‘g‘ri davrda 422; job `reports` navbatiga tushadi.

✓ `GenerateSalesReport` da `tries`, `timeout` (< retry_after), `backoff` va `failed()` bor.

✓ Hisobot fayli foydalanuvchi papkasida; begona foydalanuvchi uni yuklab ololmaydi.

✓ `ProductRepository::lowStock()` mavjud qoldiqni (`quantity – reserved`) bitta SQL so‘rov bilan hisoblaydi.

✓ `SendLowStockAlert` idempotent: ikki marta ishga tushirilsa ham bir kunda bitta xat.

✓ `LowStockNotification` mail va database kanallariga yuboradi, massiv qabul qiladi.

✓ `php artisan schedule:list` da vazifa `0 9 * * 1-5` va Asia/Tashkent bilan ko‘rinadi.

📤 GitHub repo dagi `feature/queues` tarmog‘idan ochilgan Pull Request havolasini yuboring; PR tavsifiga Mailpit dagi xat, worker logi va `schedule:list` natijasining skrinshotlarini qo‘shing.

Telefon kamerasi bilan skanerlang

Loyiha havolasini amaliy sahifasidagi «Loyiha bosqichi» bo‘limida topshiring.

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Oson · 10 ball

4-topshiriq: Kam qoldiqdagi mahsulotlar

`SendLowStockAlert` uchun kam qoldiqni hisoblang. `$qoldiqlar` — omborlar bo‘yicha qatorlar: `['sku' => string, 'quantity' => int, 'reserved' => int]`; bitta mahsulot bir nechta omborda bo‘lishi mumkin. `$minimal` — `sku => min_stock`. Mahsulotning mavjud qoldig‘i = barcha omborlardagi `quantity - reserved` yig‘indisi; qoldiq qatori umuman bo‘lmasa — 0. Mavjud qoldiq `min_stock` dan **qat’iy kichik** bo‘lgan mahsulotlarni qaytaring. `$minimal` da yo‘q SKU lar e’tiborga olinmaydi.

Imzo: `function kam_qoldiq(array $qoldiqlar, array $minimal): array`

Qaytaradi: `list` — har biri `['sku' => string, 'available' => int, 'min' => int]`, `sku` bo‘yicha o‘sish tartibida.

MISOL · CHAQIRUV

```
kam_qoldiq([['sku' => 'UN-001', 'quantity' => 10, 'reserved' => 8], ['sku' => 'UN-001', 'quantity' => 1, 'reserved' => 0], ['sku' => 'SH-002', 'quantity' => 50, 'reserved' => 5]], ['UN-001' => 5, 'SH-002' => 10])
```

MISOL · NATIJA

```
[{"sku":"UN-001","available":3,"min":5}]
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · O‘rta · 20 ball

5-topshiriq: Kam qoldiq xabari matni

`LowStockNotification::toMail()` uchun xabar tuzing. `$items` — `['sku' => string, 'available' => int, 'min' => int]` ro‘yxati. Mavzu: `Kam qoldiq: N ta mahsulot`. Qatorlar `SKU: available / min` ko‘rinishida, kamomad (`min − available`) bo‘yicha kamayish tartibida, teng bo‘lsa `sku` bo‘yicha o‘sish tartibida; faqat birinchi `$limit` tasi. `more` — ko‘rsatilmay qolganlar soni. Ro‘yxat bo‘sh bo‘lsa xabar yuborilmaydi — `DomainException`; `$limit < 1` bo‘lsa — `InvalidArgumentException`.

Imzo: `function kam_qoldiq_xabari(array $items, int $limit = 3): array`

Qaytaradi: `['subject' => string, 'lines' => list<string>, 'more' => int]` — kalitlar shu tartibda.

MISOL · CHAQIRUV

```
kam_qoldiq_xabari([['sku' => 'UN-001', 'available' => 4, 'min' => 10], ['sku' => 'SH-002', 'available' => 0, 'min' => 5]])
```

MISOL · NATIJA

```
{"subject":"Kam qoldiq: 2 ta mahsulot","lines":["UN-001: 4 / 10","SH-002: 0 / 5"],"more":0}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · O‘rta · 20 ball

6-topshiriq: Idempotentlik belgisi (Cache::add modeli)

`Cache::add` ga o‘xshash ombor yozing. `add($key, $now, $ttl)` — kalit yo‘q yoki muddati o‘tgan bo‘lsa, uni `$now + $ttl` gacha saqlaydi va `true` qaytaradi; kalit hali amalda bo‘lsa — hech narsa o‘zgartirmay `false`. Kalit `$now < muddat` bo‘lganda amalda hisoblanadi. `has($key, $now)` — kalit amaldami. `forget($key)` — kalitni o‘chiradi. `$ttl < 1` bo‘lsa — `InvalidArgumentException`. Vaqt — soniyalardagi butun son (test uchun tashqaridan beriladi).

Imzo: `final class IdempotencyStore`

Qaytaradi: `add` — `bool`, `has` — `bool`, `forget` — `void`. Testlar natijalarni massivga yig‘ib tekshiradi.

MISOL · CHAQIRUV

```
(function () { $s = new IdempotencyStore(); return [$s->add('low-stock:2026-10-27', 100, 60), $s->add('low-stock:2026-10-27', 120, 60), $s->has('low-stock:2026-10-27', 159)]; })()
```

MISOL · NATIJA

```
[true,false,true]
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Qiyin · 30 ball

7-topshiriq: Navbat simulyatori: tries va noyob joblar

`QueueSim` worker ishini modellashtiradi. Konstruktor `$tries` (jami urinishlar, < 1 bo‘lsa `InvalidArgumentException`). `push($id, $failTimes)` — jobni navbat oxiriga qo‘yadi; job birinchi `$failTimes` urinishda yiqiladi (`$failTimes < 0` — `InvalidArgumentException`). ShouldBeUnique kabi: shu `id` li job navbatda turgan bo‘lsa, `push` uni qo‘shmaydi va `false` qaytaradi, aks holda `true`. `run()` navbatni FIFO tartibda bo‘shatadi: muvaffaqiyat — logga `id#urinish:ok`; yiqilish va urinish `< tries` — `id#urinish:retry` va job navbat **oxiriga** qaytadi; yiqilish va urinish `= tries` — `id#urinish:failed` va `id` failed ro‘yxatiga.

Imzo: `final class QueueSim`

Qaytaradi: `run(): ['log' => list<string>, 'failed' => list<string>]`.

MISOL · CHAQIRUV

```
(function () { $q = new QueueSim(3); $q->push('A', 0); $q->push('B', 1); return $q->run(); })()
```

MISOL · NATIJA

```
{"log":["A#1:ok","B#1:retry","B#2:ok"],"failed":[]}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Qiyin · 30 ball

8-topshiriq: Hisobot davrini oylarga bo‘lish (Bus::batch uchun)

Katta davr hisobotini `Bus::batch` dagi parallel joblarga bo‘lish uchun davrni kalendar oylariga ajrating. `$dan` va `$gacha` — `Y-m-d` formatidagi sanalar (ikkalasi ham kiradi). Har bir bo‘lak `['from' => ..., 'to' => ...]`: birinchi bo‘lak `$dan` dan oy oxirigacha, oxirgisi oy boshidan `$gacha` gacha, o‘rtadagilar to‘liq oylar. Sana formati yoki qiymati noto‘g‘ri (masalan `2026-02-30`) — `InvalidArgumentException`; `$dan > $gacha` — `DomainException`; 12 tadan ko‘p bo‘lak — `LengthException`.

Imzo: `function hisobot_bolaklari(string $dan, string $gacha): array`

Qaytaradi: `list<array{from: string, to: string}>` — xronologik tartibda.

MISOL · CHAQIRUV

```
hisobot_bolaklari('2026-01-15', '2026-03-10')
```

MISOL · NATIJA

```
[{"from":"2026-01-15","to":"2026-01-31"},{"from":"2026-02-01","to":"2026-02-28"},{"from":"2026-03-01","to":"2026-03-10"}]
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

TOPSHIRIQLAR VA TOPSHIRISH

10-amaliy: 8 ta PHP topshiriq

1 **Navbatlar ustuvorligi**
Oson · 10 ball · ma’ruzada

2 **Qayta urinish yoki failed_jobs?**
O‘rta · 20 ball · ma’ruzada

3 **Cron ifodasi vaqtga mosmi?**
O‘rta · 20 ball · ma’ruzada

4 **Kam qoldiqdagi mahsulotlar**
Oson · 10 ball

5 **Kam qoldiq xabari matni**
O‘rta · 20 ball

6 **Idempotentlik belgisi (Cache::add modeli)**
O‘rta · 20 ball

7 **Navbat simulyatori: tries va noyob joblar**
Qiyin · 30 ball

8 **Hisobot davrini oylarga bo‘lish (Bus::batch uchun)**
Qiyin · 30 ball

Telefon kamerasi bilan skanerlang

PHP kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

Loyiha havolasini ham shu sahifada topshirasiz; o‘qituvchi baho qo‘yadi.